WHAT IS CLAIMED IS:

1. A computer-implemented method of determining if a query element is included in a set of elements, the method comprising:

building a data structure based upon information identifying elements in the set of elements;

receiving information identifying the query element; and

using the data structure to determine if the query element is included in the set of elements such that the number of comparisons needed to determine if the query element is included in the set of elements is proportional to a length of the query element and independent of the number of elements in the set of elements.

2. The method of claim 1 wherein the query element is of length “q” and at most “q” character comparisons are needed to determine if the query element is included in the set of elements.

3. The method of claim 1 wherein:

the set of elements contains elements from a domain Σ having a character set of “m” characters, wherein “Z” is the maximum possible length of an element in domain Σ and “Y” is the length of the longest element in the set of elements such that $1 \leq Y \leq Z$; and

building the data structure comprises building the data structure comprising a plurality of memory structures headed by a root memory structure, each memory structure in the plurality of memory structures comprising a first memory location and an array of “m” memory locations.

4. The method of claim 3 wherein:

the data structure comprises a total of (Y+1) levels; and

each memory structure in the data structure belongs to a level L, where $(0 \leq L \leq Y)$, the level for a particular memory structure denoting the number of memory structures, starting with the root memory structure, that have to be traversed to reach the particular memory structure, the root memory structure belonging to level 0.

5. The method of claim 4 wherein building the data structure based upon information identifying the elements in the set of elements comprises:

for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Y$, for each c_i where $1 \leq i \leq$ f, starting with i = 1:

(a) selecting a memory structure at level "(i-1)";

(b) if a memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure does not refer to another memory structure in the database, storing an address of a new memory structure at level "i" in the memory location corresponding to character c_i in the array of memory locations of the selected memory structure;

(c) selecting the memory structure at level "i" whose address is stored in the memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure;

(d) if ("i" is equal to "f"), storing a reference to element "R" in the first memory location of the memory structure selected in step (c);

(e) incrementing the value of "i" by one; and

(f) repeating steps (b), (c), (d), and (e) for each c_i where ("i" ≤ "f").

6. The method of claim 5 wherein:

receiving information identifying the query element comprises:

receiving information identifying a query element k, where $k = c_1c_2...c_q$ for some $q \leq Z$;

using the data structure to determine if the query element is included in the set of elements comprises:

for each c_i of k where $1 \leq i \leq$ f, starting with i = 1:

(a) selecting a memory structure of the database at level "(i-1)";

(b) if a memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure does not refer to another memory structure in the database, outputting a signal indicating that the query element is not included in the set of elements;

(c) if the memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure stores an address of a memory structure of the database at level "i", selecting the memory structure at level "i" whose address is stored;

(d) incrementing the value of "i" by one; and

(e) repeating steps (b), (c), and (d) while (“i” ≤ “q”) and the signal indicating that the query element is not included in the set of elements has not been output; and

if the signal indicating that the query element is not included in the set of elements has not been output:

determining if the first memory location of the memory structure selected in step (c) refers to the query element; and

if the first memory location of the memory structure selected in step (c) refers to the query element, outputting a signal indicating that the query element is included in the set of elements, else outputting a signal indicating that the query element is not included in the set of elements.

7. The method of claim 3 wherein building the data structure based upon information identifying the elements in the set of elements comprises:

for each element “R” in the set of elements, where $R = c_1 c_2 \ldots c_f$ for some $f \leq Y$, where each character c_i belongs to the character set of domain Σ, and $1 \leq i \leq f$, storing information in the database indicating the position and identity of each character in element R.

8. The method of claim 7 wherein using the data structure to determine if the query element is included in the set of elements comprises:

determining if the query element is included in the set of elements based upon information stored by the database and information identifying characters and their positions in the query element.

9. The method of claim 3 wherein building the data structure based upon information identifying the elements in the set of elements comprises:

for each element “R” in the set of elements:

(a) selecting the root memory structure of the data structure as the selected memory structure;

(b) selecting the first character of element R;

(c) if a memory location corresponding to the selected character in the array of memory locations of the selected memory structure does not refer to another memory structure in the data structure, storing an address of a new memory structure in the memory location corresponding to the selected character in the array of memory locations of the presently selected memory structure;

(d) selecting the memory structure whose address is stored in the memory location corresponding to the selected character in the array of memory locations of the selected memory structure as the selected memory structure; and

(e) if the selected character is the last character of element R, storing a reference to element R in the first memory location of the memory structure selected in step (d),

else, selecting the next character of element R, and repeating steps (c), (d), and (e).

10. The method of claim 9 wherein using the data structure to determine if the query element is included in the set of elements comprises:

(a) selecting the root memory structure of the data structure as the selected memory structure;

(b) selecting the first character of the query element;

(c) if a memory location corresponding to the selected character in the array of memory locations of the selected memory structure does not refer to another memory structure in the data structure, outputting a signal indicating that the query element is not included in the set of elements,

else, selecting the memory structure whose address is stored as the selected memory element; and

(d) if the selected character is the last character of the query element:

determining if the first memory location of the memory structure selected in step (c) refers to the query element; and

if the first memory location of the memory structure selected in step (c) refers to the query element, outputting a signal indicating that the query element is included in the set of elements, else outputting a signal indicating that the query element is not included in the set of elements;

else:

selecting the next character of the query element , and repeating steps (c) and (d).

11. The method of claim 1 wherein a size of the data structure is independent of the number of elements in the set of elements.

12. The method of claim 11 wherein:

the set of elements contains elements from a domain Σ having a character set of "m" characters, and wherein "Z" is the maximum possible length of an element in domain Σ; and

the data structure comprises "Z" memory structures, each memory structure comprising "m" slots, each slot comprising a first memory location and an array of memory locations, each array of memory locations comprising "(m+1)" memory locations.

13. The method of claim 12 wherein building the data structure based upon information identifying the elements in the set of elements comprises:

initializing the first memory location and memory locations in the array of memory locations of each slot in each memory structure to null values;

for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Z$, for each c_i where $1 \leq i \leq$ f:

if ("i" < "f"):

storing a non-null value in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i; and

if ("i" is equal to "f"):

storing a non-null value in the $(m+1)^{th}$ memory location of the array of memory locations of the slot corresponding to c_i of memory structure i; and

storing a reference to element "R" in the first memory location of the slot corresponding to c_i of memory structure i.

14. The method of claim 13 wherein:

receiving information identifying the query element comprises:

receiving information identifying a query element "k", where $k = c_1c_2...c_q$ for some $q \leq Z$;

using the data structure to determine if the query element is included in the set of elements comprises:

outputting a signal indicating that the query element is included in the set of elements if, for each c_i of k:

if ("i" < "q"), a non-null value is stored in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i; and

if ("i" is equal to "q"), a non-null value is stored in the $(m+1)^{th}$ memory location in the array of memory locations of the slot corresponding to c_i of memory structure i, and the first memory location of the slot corresponding to c_i of memory structure i refers to the query element.

15. The method of claim 13 wherein:

receiving information identifying the query element comprises:

receiving information identifying a query element "k", where $k = c_1 c_2 ... c_q$ for some $q \le Z$; and

using the data structure to determine if the query element is included in the set of elements comprises:

outputting a signal indicating that the query element is not included in the set of elements if, for any c_i of k:

if ("i" is equal to "q"), a null value is stored in the $(m+1)^{th}$ memory location in the array of memory locations of the slot corresponding to c_i of memory structure i, or the first memory location of the slot corresponding to c_i of memory structure i does not refer to the query element; and

if ("i" < "q"), a null value is stored in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i.

16. A system for determining if a query element is included in a set of elements, the system comprising:

a processor;

a memory coupled to the processor, the memory configured to store a plurality of code modules executable by the processor, the plurality of code modules comprising:

a code module for building a data structure based upon information identifying elements in the set of elements;

a code module for receiving information identifying the query element; and

a code module for using the data structure to determine if the query element is included in the set of elements such that the number of comparisons needed to determine if the query element is included in the set of elements is proportional to a length of the query element and independent of the number of elements in the set of elements.

17. The system of claim 16 wherein the query element is of length "q" and at most "q" character comparisons are needed to determine if the query element is included in the set of elements.

18. The system of claim 16 wherein:

the set of elements contains elements from a domain Σ having a character set of "m" characters, wherein "Z" is the maximum possible length of an element in domain Σ and "Y" is the length of the longest element in the set of elements such that $1 \leq Y \leq Z$; and

the code module for building the data structure comprises a code module for building the data structure comprising a plurality of memory structures headed by a root memory structure, each memory structure in the plurality of memory structures comprising a first memory location and an array of "m" memory locations.

19. The system of claim 18 wherein:

the data structure comprises a total of (Y+1) levels; and

each memory structure in the data structure belongs to a level L, where $(0 \leq L \leq Y)$, the level for a particular memory structure denoting the number of memory structures, starting with the root memory structure, that have to be traversed to reach the particular memory structure, the root memory structure belonging to level 0.

20. The system of claim 19 wherein the code module for building the data structure based upon information identifying the elements in the set of elements comprises:

for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Y$, for each c_i where $1 \leq i \leq$ f, starting with i = 1:

(a) a code module for selecting a memory structure at level "(i-1)";

(b) if a memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure does not refer to another memory structure in the database, a code module for storing an address of a new memory structure at level "i" in the memory location corresponding to character c_i in the array of memory locations of the selected memory structure;

(c) a code module for selecting the memory structure at level "i" whose address is stored in the memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure;

(d) if ("i" is equal to "f"), a code module for storing a reference to element "R" in the first memory location of the memory structure selected in step (c);

(e) a code module for incrementing the value of "i" by one; and

(f) a code module for repeating steps (b), (c), (d), and (e) for each c_i where ("i" ≤ "f").

21. The system of claim 20 wherein:

the code module for receiving information identifying the query element comprises:

a code module for receiving information identifying a query element k, where $k = c_1c_2...c_q$ for some $q \le Z$;

the code module for using the data structure to determine if the query element is included in the set of elements comprises:

for each c_i of k where $1 \le i \le$ f, starting with i = 1:

(a) a code module for selecting a memory structure of the database at level "(i-1)";

(b) if a memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure does not refer to another memory structure in the database, a code module for outputting a signal indicating that the query element is not included in the set of elements;

(c) if the memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure stores an address of a memory structure of the database at level "i", a code module for selecting the memory structure at level "i" whose address is stored;

(d) a code module for incrementing the value of "i" by one; and

(e) a code module for repeating steps (b), (c), and (d) while ("i" ≤ "q") and the signal indicating that the query element is not included in the set of elements has not been output; and if the signal indicating that the query element is not included in the set of elements has not been output:

a code module for determining if the first memory location of the memory structure selected in step (c) refers to the query element; and

if the first memory location of the memory structure selected in step (c) refers to the query element, a code module for outputting a signal indicating that the query element is included in the set of elements, else a code module for outputting a signal indicating that the query element is not included in the set of elements.

22. The system of claim 18 wherein the code module for building the data structure based upon information identifying the elements in the set of elements comprises:

for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Y$, where each character c_i belongs to the character set of domain Σ, and $1 \leq i \leq f$, a code module for storing information in the database indicating the position and identity of each character in element R.

23. The system of claim 22 wherein the code module for using the data structure to determine if the query element is included in the set of elements comprises:

a code module for determining if the query element is included in the set of elements based upon information stored by the database and information identifying characters and their positions in the query element.

24. The system of claim 18 wherein the code module for building the data structure based upon information identifying the elements in the set of elements comprises:

for each element "R" in the set of elements:

(a) a code module for selecting the root memory structure of the data structure as the selected memory structure;

(b) a code module for selecting the first character of element R;

(c) if a memory location corresponding to the selected character in the array of memory locations of the selected memory structure does not refer to another memory structure in the data structure, a code module for storing an address of a new memory structure in the memory location corresponding to the selected character in the array of memory locations of the presently selected memory structure;

(d) a code module for selecting the memory structure whose address is stored in the memory location corresponding to the selected character in the array of memory locations of the selected memory structure as the selected memory structure; and

(e) if the selected character is the last character of element R, a code module for storing a reference to element R in the first memory location of the memory structure selected in step (d),

else, a code module for selecting the next character of element R, and repeating steps (c), (d), and (e).

25. The system of claim 24 wherein the code module for using the data structure to determine if the query element is included in the set of elements comprises:

(a) a code module for selecting the root memory structure of the data structure as the selected memory structure;

(b) a code module for selecting the first character of the query element;

(c) if a memory location corresponding to the selected character in the array of memory locations of the selected memory structure does not refer to another memory structure in the data structure, a code module for outputting a signal indicating that the query element is not included in the set of elements,

else, a code module for selecting the memory structure whose address is stored as the selected memory element; and

(d) if the selected character is the last character of the query element:

a code module for determining if the first memory location of the memory structure selected in step (c) refers to the query element; and

if the first memory location of the memory structure selected in step (c) refers to the query element, a code module for outputting a signal indicating that the query element is included in the set of elements, else a code module for outputting a signal indicating that the query element is not included in the set of elements;

else:

a code module for selecting the next character of the query element , and repeating steps (c) and (d).

26. The system of claim 16 wherein a size of the data structure is independent of the number of elements in the set of elements.

27. The system of claim 26 wherein:

the set of elements contains elements from a domain Σ having a character set of "m" characters, and wherein "Z" is the maximum possible length of an element in domain Σ; and

the data structure comprises "Z" memory structures, each memory structure comprising "m" slots, each slot comprising a first memory location and an array of memory locations, each array of memory locations comprising "(m+1)" memory locations.

28. The system of claim 27 wherein the code module for building the data structure based upon information identifying the elements in the set of elements comprises:

a code module for initializing the first memory location and memory locations in the array of memory locations of each slot in each memory structure to null values;

for each element "R" in the set of elements, where $R = c_1 c_2 \ldots c_f$ for some $f \leq Z$, for each c_i where $1 \leq i \leq$ f:

if ("i" < "f"):

a code module for storing a non-null value in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i; and

if ("i" is equal to "f"):

a code module for storing a non-null value in the $(m+1)^{th}$ memory location of the array of memory locations of the slot corresponding to c_i of memory structure i; and

a code module for storing a reference to element "R" in the first memory location of the slot corresponding to c_i of memory structure i.

29. The system of claim 28 wherein:

the code module for receiving information identifying the query element comprises:

a code module for receiving information identifying a query element "k", where $k = c_1 c_2 \ldots c_q$ for some $q \leq Z$;

the code module for using the data structure to determine if the query element is included in the set of elements comprises:

a code module for outputting a signal indicating that the query element is included in the set of elements if, for each c_i of k:

if (“i” < “q”), a non-null value is stored in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i; and

if (“i” is equal to “q”), a non-null value is stored in the $(m+1)^{th}$ memory location in the array of memory locations of the slot corresponding to c_i of memory structure i, and the first memory location of the slot corresponding to c_i of memory structure i refers to the query element.

30. The system of claim 28 wherein:

the code module for receiving information identifying the query element comprises:

a code module for receiving information identifying a query element “k”, where $k = c_1 c_2 \ldots c_q$ for some $q \le Z$; and

the code module for using the data structure to determine if the query element is included in the set of elements comprises:

a code module for outputting a signal indicating that the query element is not included in the set of elements if, for any c_i of k:

if (“i” is equal to “q”), a null value is stored in the $(m+1)^{th}$ memory location in the array of memory locations of the slot corresponding to c_i of memory structure i, or the first memory location of the slot corresponding to c_i of memory structure i does not refer to the query element; and

if (“i” < “q”), a null value is stored in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i.

31. A computer program product stored on a computer-readable storage medium for determining if a query element is included in a set of elements, the computer program product comprising:

code for building a data structure based upon information identifying elements in the set of elements;

code for receiving information identifying the query element; and

code for using the data structure to determine if the query element is included in the set of elements such that the number of comparisons needed to determine if the query element is included in the set of elements is proportional to a length of the query element and independent of the number of elements in the set of elements.

32. The computer program product of claim 31 wherein the query element is of length "q" and at most "q" character comparisons are needed to determine if the query element is included in the set of elements.

33. The computer program product of claim 31 wherein:

the set of elements contains elements from a domain Σ having a character set of "m" characters, wherein "Z" is the maximum possible length of an element in domain Σ and "Y" is the length of the longest element in the set of elements such that $1 \leq Y \leq Z$; and

the code for building the data structure comprises code for building the data structure comprising a plurality of memory structures headed by a root memory structure, each memory structure in the plurality of memory structures comprising a first memory location and an array of "m" memory locations.

34. The computer program product of claim 33 wherein:

the data structure comprises a total of (Y+1) levels; and

each memory structure in the data structure belongs to a level L, where ($0 \leq L \leq Y$), the level for a particular memory structure denoting the number of memory structures, starting with the root memory structure, that have to be traversed to reach the particular memory structure, the root memory structure belonging to level 0.

35. The computer program product of claim 33 wherein the code for building the data structure based upon information identifying the elements in the set of elements comprises:

for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Y$, where each character c_i belongs to the character set of domain Σ, and $1 \leq i \leq f$, code for storing information in the database indicating the position and identity of each character in element R.

36. The computer program product of claim 35 wherein the code for using the data structure to determine if the query element is included in the set of elements comprises:

code for determining if the query element is included in the set of elements based upon information stored by the database and information identifying characters and their positions in the query element.

37. The computer program product of claim 31 wherein:

a size of the data structure is independent of the number of elements in the set of elements;

the set of elements contains elements from a domain Σ having a character set of "m" characters, and wherein "Z" is the maximum possible length of an element in domain Σ; and

the data structure comprises "Z" memory structures, each memory structure comprising "m" slots, each slot comprising a first memory location and an array of memory locations, each array of memory locations comprising "(m+1)" memory locations.

38. The computer program product of claim 37 wherein the code for building the data structure based upon information identifying the elements in the set of elements comprises:

code for initializing the first memory location and memory locations in the array of memory locations of each slot in each memory structure to null values;

for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Z$, for each c_i where $1 \leq i \leq$ f:

if ("i" < "f"):

code for storing a non-null value in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i; and

if ("i" is equal to "f"):

code for storing a non-null value in the $(m+1)^{th}$ memory location of the array of memory locations of the slot corresponding to c_i of memory structure i; and

code for storing a reference to element "R" in the first memory location of the slot corresponding to c_i of memory structure i.

39. The computer program product of claim 38 wherein:

the code for receiving information identifying the query element comprises:

code for receiving information identifying a query element "k", where $k = c_1c_2...c_q$ for some $q \leq Z$;

the code for using the data structure to determine if the query element is included in the set of elements comprises:

code for outputting a signal indicating that the query element is included in the set of elements if, for each c_i of k:

if ("i" < "q"), a non-null value is stored in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i; and

if ("i" is equal to "q"), a non-null value is stored in the $(m+1)^{th}$ memory location in the array of memory locations of the slot corresponding to c_i of memory structure i, and the first memory location of the slot corresponding to c_i of memory structure i refers to the query element.

40. The computer program product of claim 38 wherein:

the code for receiving information identifying the query element comprises:

code for receiving information identifying a query element "k", where $k = c_1c_2...c_q$ for some $q \leq Z$;

the code for using the data structure to determine if the query element is included in the set of elements comprises:

code for outputting a signal indicating that the query element is not included in the set of elements if, for any c_i of k:

if ("i" is equal to "q"), a null value is stored in the $(m+1)^{th}$ memory location in the array of memory locations of the slot corresponding to c_i of memory structure i, or the first memory location of the slot corresponding to c_i of memory structure i does not refer to the query element; and

if ("i" < "q"), a null value is stored in a memory location corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory structure i.